

Dartmouth College

Dartmouth Digital Commons

Computer Science Technical Reports

Computer Science

8-1-2018

Battery-Free Eye Tracker on Glasses

Tianxing Li

Dartmouth College

Xia Zhou

Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/cs_tr



Part of the [Computer Sciences Commons](#)

Dartmouth Digital Commons Citation

Li, Tianxing and Zhou, Xia, "Battery-Free Eye Tracker on Glasses" (2018). Computer Science Technical Report TR2018-861. https://digitalcommons.dartmouth.edu/cs_tr/378

This Technical Report is brought to you for free and open access by the Computer Science at Dartmouth Digital Commons. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

Battery-Free Eye Tracker on Glasses

Tianxing Li and Xia Zhou

Department of Computer Science, Dartmouth College, Hanover, NH

{tianxing,xia}@cs.dartmouth.edu

*Dartmouth College Computer Science Technical Report TR2018-861**

ABSTRACT

This paper presents a battery-free wearable eye tracker that tracks both the 2D position and diameter of pupil based on pupil's light absorption property. With a few near-infrared (NIR) lights and photodiodes around the eye, NIR lights sequentially illuminate the eye from various directions while photodiodes sense spatial patterns of reflected light, which are used to infer pupil's position and diameter on the fly via a lightweight inference algorithm. The system also exploits characteristics of different eye movement stages and adjusts its sensing and computation accordingly for further energy savings. A prototype is built with off-the-shelf hardware components and integrated into a regular pair of glasses. Experiments with 22 participants show that the system achieves 0.8-mm mean error in tracking pupil position (2.3 mm at the 95th percentile) and 0.3-mm mean error in tracking pupil diameter (0.9 mm at the 95th percentile) at 120-Hz output frame rate, consuming $395\mu\text{W}$ mean power supplied by two small, thin solar cells on glasses side arms.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile devices**; • **Computer systems organization** → *Sensors and actuators*;

KEYWORDS

Eye tracking, light sensing, energy harvesting

1 INTRODUCTION

Eye movement is a vital biological marker. Continuous eye tracking is critical for understanding its correlation with cognitive processes [43, 46, 63], identifying health issues (e.g., mental disorders, cognitive dysfunctions) [35, 37, 53], and assessing the effectiveness of clinical treatments [58]. It is also crucial for the development of human-to-computer interaction by allowing hands-free, attentive user interfaces, and the apprehension of user affective states [62].

Supporting above applications requires eye tracking at the sub-millimeter level with high tracking rates, because eye movement manifests rapid (e.g., $500^\circ/\text{s}$ or 25-cm/s) scanning, referred to as *saccades*, mixed with eye fixations (\approx

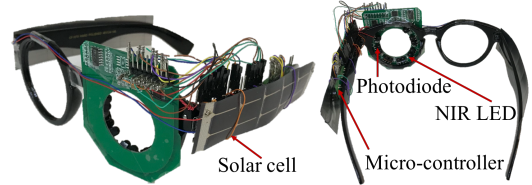


Figure 1: Integrating our eye tracker into a regular pair of glasses. The system relies on NIR lights and photodiodes for eye tracking and is powered by two small, thin solar cells on the side arms of the glasses.

200 milliseconds on average). Real-time, accurate measures of these eye movement stages are essential. As examples, the anti-saccade task is a gross estimation of injury or dysfunction of the frontal lobe, which helps measure the risks of schizophrenia. This task requires eye tracking with less than $3^\circ/1.5\text{-mm}$ error and at least 120 Hz to extract saccadic eye movement [53]; when tracking user's attention in augmented-reality glasses (e.g., Microsoft HoloLens), $2^\circ/1\text{-mm}$ tracking error maps to 3.5-cm deviation on a object one meter away from the user [1]. Additionally, to support long-term usage, the eye tracker needs to be portable and low-power, eliminating the need of frequent charging.

Existing wearable eye trackers still fall short in achieving these goals, mainly because of the inherent conflict between tracking performance and energy consumption. The common approach uses cameras to capture eye images and applies intensive image processing to identify pupil position and size. As a result, high-end eye trackers entail a prohibitive cost (e.g., \$15K+ for a Tobii wearable eye tracker [7]), and often require external battery packs that can be cumbersome to carry for day-to-day eye monitoring. Recent studies [44, 45, 77] have lowered the energy consumption of camera-based wearable eye trackers, with the latest design [45] projected to consume tens of milliwatts at tracking rates above 100 Hz. A recent work [41] examines eye tracking without cameras for virtual reality, further lowering energy consumption, yet at the cost of much higher tracking error ($10^\circ/5\text{-mm}$ mean error).

We set out to further push the envelope in this work. We seek a low-cost design of wearable eye tracker that can operate at high tracking rates (above 100 Hz) with sub-millimeter accuracy while consuming power at microwatt levels. With the overall power consumption hundreds of times lower than the state-of-the-art [44, 45, 77], the eye tracker can ultimately

*This technical report is an extended version of a MobiCom 2018 paper with the same title [42].

be powered by energy harvested from ambient environment (e.g., indoor lighting) and thus require no battery. It eases the integration of the eye tracker into a regular pair of glasses without any battery source (Figure 1).

We achieve these goals with two key design elements. First, like [41], our approach is *camera-less* at the core, leveraging low-cost (e.g., \$1), small photodiodes on the lens frame to sense light reflected by the eyeball. The reflected light is used to infer the pupil position based on pupil’s light absorption property [74]. Our approach differs from [41] in that it extends the application scenario to regular glasses and augmented reality glasses, by adding *multiple* small (e.g., 5 mm²) near-infrared (NIR) light emitting diodes (LEDs) as controlled light sources around the eye. NIR LEDs, as point lights, in turn emit ultra-short light pulses onto the eye from different directions, while photodiodes sense reflected NIR light from their vantage points. We optimize the layout and tilt angles of LEDs and photodiodes to mitigate the interference of eyelashes. Aggregating spatial patterns of reflected light under different NIR lights allow us to boost the accuracy of inferring pupil’s position and diameter with a lightweight inference algorithm based on supervised learning.

The second element and main contribution of this work is the design principle of *adapting eye tracker’s sensing and computation based on current eye movement characteristics*. Prior studies [19] reveal that pupil either is almost stationary (eye fixation), or follows a path with relatively constant speed (smooth pursuit), or moves rapidly with little predictability (saccades), or is covered by the eyelid during a blink. It implies that eye movement is predictable in most cases except saccades. During stages (e.g., fixation, smooth pursuit) with predictable movement, this predictability can be exploited to adjust the frequency of light sensing and pupil inference for significant energy savings without sacrificing tracking accuracy. It is particularly beneficial for eye tracking at high rates (above 100 Hz), since each eye movement type/stage can last hundreds of milliseconds [19, 61], corresponding to at least tens of inference results within each stage.

The challenge of realizing this design principle is to timely detect transitions of movement stages so that sensing and computation are adapted correctly. Additionally, we need to avoid error propagation as we exploit movement predictability for pupil inferences. Inspired by prior works on eye movement detection and analysis, we consider pupil movement trajectory, velocity, and acceleration within a small time window to detect movement stages. We compare these metrics to adaptive thresholds using the constant false alarm rate (CFAR) algorithm [66] to ensure robust detection against varying noise levels. We adjust the rates of sensing and inference at multiple levels for immediate detection of stage transition. We also interleave predictive inferences with full-fledged inferences that clear accumulated errors periodically.

We design and fabricate a prototype with off-the-shelf hardware. The prototype consists of a thin (0.8-mm) customized printed-circuit board (PCB) hosting 6 NIR LEDs and 12 photodiodes (Figure 8(a)), a micro-controller (MSP432), and an energy harvester with two thin small solar cells attached to the side arms of the glasses. We integrate it to a regular pair of glasses and test it with 22 participants across a variety of activities. Our key findings are as below:

- Our system achieves 0.8-mm mean accuracy (with 2.3 mm at the 95th percentile) in tracking pupil’s 2D position, and 0.3-mm mean accuracy (with 0.9 mm at the 95th percentile) in tracking pupil’s diameter;
- Eye movement stages (blinks, fixations, saccades) are detected with 92.1% precision and 91.9% recall across users, with the F1 score of 0.92;
- The system produces a pupil inference result in position and size within 600 μ s, potentially supporting tracking rates above 1.7 KHz;
- With efficient adaptation, the system consumes 395 μ W on average across various user activities while supporting 120-Hz output frame rate, and thus can be powered by two thin solar cells on the side arms of the glasses.

2 METHODOLOGY AND CHALLENGES

The rationale of our camera-less eye tracking stems from pupil’s light absorption effect when the light source and optical receiver are not aligned in the optical axis. As a hole in the center of the iris, pupil allows light rays to enter the eye and strike the retina. Acting as a retroreflector, retina reflects light back to the light source [74], leaving a receiver off the optical axis of the light source perceiving no reflected light from the pupil. This property has been exploited by past works [26, 47] that seek a dark pupil in captured images.

We exploit this effect for eye tracking without cameras. Specifically we consider the use of small photodiodes around the eye to sense light reflected by the eyeball and capture changes in reflected light caused by pupil movement. A recent study [41] applied a similar methodology in the context of virtual reality (VR), where a dual-sided light sensing unit on the VR lens senses incoming screen light and reflected light from the eye. This design handles incoming light variations by modeling the relationship between sensed incoming light and reflected light. The calibration of the model is possible in VR, where VR screen is the sole light source in a fixed direction and can play judiciously-designed VR content to shorten the calibration for a user. Applying this design for general glasses, however, is infeasible, given the uncontrolled, diverse nature of ambient light sources. Front sensors cannot provide representative samples of ambient light coming in unknown directions. Also, calibrating the model in diverse ambient light conditions entails heavy overhead.

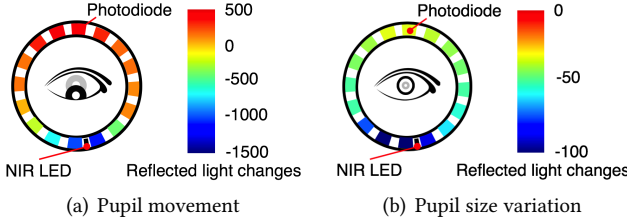


Figure 2: Examining pupil’s light absorption effect with a NIR LED and 16 photodiodes around the eye. (a)-(b) show changes in reflected NIR light (ADC readings directly related to the NIR power) when the pupil moves from center to bottom (a) and a center pupil dilates (b).

To address this problem, we consider sensing with NIR light, given that ambient light contains much weaker energy in NIR than visible light. We add an NIR LED as a controlled, point light source near the eye. It emits a low-power¹, imperceptible NIR light beam onto the eye in a known direction, while a set of NIR photodiodes circling the eye sense NIR light reflected by the eyeball after both specular and diffuse reflections. Each photodiode has a limited field-of-view and thus perceives the light reflected only by a small region of the eye. As the pupil moves, it weakens the light reflected in its direction; as the pupil narrows or widens during constriction or dilation, it absorbs less or more light, resulting into a global rise or fall of reflected light. Thus, pupil movement and pupillary response result in changes in reflected light, providing the basis for tracking pupil’s position and size.

Experimental Validation. To examine the effect of pupil’s light absorption in the glasses context, we conduct experiments with off-the-shelf NIR LED and photodiodes. We fabricate a PCB hosting 16 photodiodes (BPV22F) and 1 NIR LED (VSMY2943)² arranged in a 3.2-cm circle. The LED is placed at the bottom and transmits at 0.1 mW/sr following the eye-safety standard [28]. The PCB also contains four amplifiers connecting to photodiodes. We attach the board to the front of the left lens of a regular pair of glasses and connect the board to a micro-controller (MSP432 w/ 14-bit ADC) that samples data from photodiodes at 200 KHz.

Figure 2 illustrates the reflected NIR light changes upon pupil movement or pupil size variation (e.g., dilation). We see that as the pupil moves from the center to bottom, bottom/top photodiodes perceive declines/increases in NIR light intensity because a bottom pupil absorbs more light rays. Pupil dilation, on the other hand, leads to the global decline of reflected light intensity as the pupil absorbs more light.

We further examine the impact of ambient light, since ambient light sources also emit energy in the NIR spectrum. In the experiment, we switch off the NIR LED in the PCB and collect photodiode data with a user wearing the prototype

under nine ambient light settings (Table 6). In all settings except direct sunlight, photodiode readings are less than 150, which is close to the noise level given that the maximum ADC value at the micro-controller is 12400 under the 2.5-V reference voltage. It indicates that indoor ambient NIR light has a negligible impact on the system. In §6.6, we will conduct detailed experiments examining the final eye-tracking performance under various ambient light conditions.

Challenges. To achieve high-performance eye tracking using the above methodology, we face two main challenges. *First*, unlike an image sensor or camera, a photodiode does not provide any spatial resolution within its field-of-view, rather, only a combined light intensity. Thus, a sub-millimeter pupil movement or size variation can result in negligible differences in reflected light intensities sensed by the photodiodes, which severely limits the system’s sensing resolution and accuracy. Furthermore, movement of other components (e.g., eyelashes, eyelids) in the eye area also affects reflected NIR light and interferes with eye tracking. *Second*, even though photodiodes consume low power, realizing high tracking rates with microwatt-level power consumption is still challenging. Existing methods commonly reduce the sleep duration to achieve high tracking rates [41, 44, 45]. This methodology significantly increases the power consumption to at least a few milliwatts, which is orders of magnitude higher than the amount of power one can harvest from ambient environment (e.g., radio signals, light, thermal or kinetic energy) [73]. An alternative method is to interpolate pupil positions based on samples under a low tracking rate. For rapid eye movement during saccades, however, such simple interpolation can cause tracking errors up to 10 mm [15].

In the next two sections, we will describe our solutions to addressing these two challenges.

3 MULTI-LIGHT EYE TRACKING

Our first design element comprises designs of both the sensing hardware and the inference algorithm to achieve sub-millimeter tracking accuracy. At the high level, we propose to combine multiple NIR lights and an array of photodiodes. NIR lights are sequentially switched on (with 10- μ s switching delay) to emit a short (e.g., 60- μ s), directional light beam, while photodiodes sense the spatial pattern of changes in reflected light under each NIR light. We then aggregate these sensing data, extract features, and feed them to a lightweight regression algorithm to infer pupil’s position and diameter on the fly. We next describe each component in detail.

3.1 Sensing with Multiple NIR Lights

We aim to boost system’s spatial sensing resolution so that it is capable of differentiating sub-millimeter pupil movements and size variations. A straightforward method is to

¹NIR irradiance power at or below 0.96 mW/cm² (i.e., 0.42 mW/sr) at eye surface is considered eye-safe [28].

²LEDs and photodiodes have peak spectrum sensitivity at 940 nm.

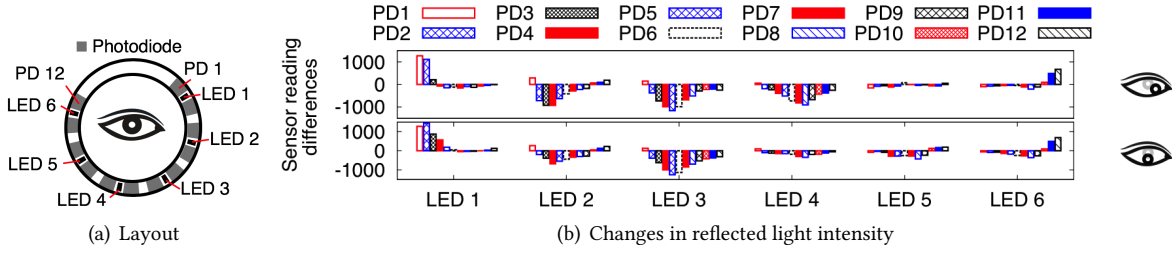


Figure 3: Sensing with multiple NIR lights around the eye. (a) shows the layout of 6 NIR LEDs and 12 photodiodes around the left eye, viewing from the back of glasses lens. We do not place LEDs or photodiodes near eye’s upper lateral canthus corner to mitigate inferences from eyelashes. (b) shows changes in reflected light caused by pupil movement when each LED is individually on.

increase the number of photodiodes while shrinking each photodiode’s field-of-view to reduce its spatial ambiguity. The sensing regions of these photodiodes jointly cover the eye surface and can be coupled with one NIR light illuminating the eye. With N photodiodes, this method provides N data points at each time instance for pupil inference. Thus, it requires a large number of photodiodes to achieve fine-grained tracking, raising concerns on both the eye tracker’s form factor and its power consumption on sensing.

To gather sensing data with a minimal number of hardware elements, we couple M NIR lights³ with N photodiodes, providing $M \times N$ data points with $(M + N)$ hardware components at each time instance. NIR lights and photodiodes circle the eye along the boundary of the glasses lens. We judiciously design their layout considering the asymmetry of the eye area. Specifically, we observe that eyelashes also reflect NIR light and interfere with our sensing. Thus, we do not place photodiodes at the top corner. We also do not place NIR light near the lateral canthus corner, as it is further from the eyeball and light rays from this direction mostly illuminate the corner skin. Figure 3(a) illustrates the layout of NIR lights and photodiodes. We slightly tilt the NIR lights and photodiodes so that they better face the eyeball. In §6.6, we will evaluate the impact of the number and layout of LEDs and photodiodes on eye-tracking performance.

We regulate NIR lights and photodiodes as follows. NIR lights in turn emit ultra-short, directional light beams from complementary directions onto the eye. Under each NIR light, photodiodes sense reflected light at their vantage points, gathering a spatial pattern of changes in reflected light. By separating NIR lights in the time domain, for a given pupil status, we obtain separate snapshots of reflected light patterns under different NIR lights. Two pupil statuses may lead to similar spatial light patterns under one light while exhibiting different patterns under another light. The combination of these patterns/snapshots refines sensing resolution and reduces spatial ambiguity. As an example, Figure 3(b) compares spatial patterns of reflected light changes caused by a

pupil moving from the center to two positions at the bottom under each light. Given the proximity of the two bottom positions, reflected light patterns are very similar under some LEDs (e.g., LED 2 and 3) while differing more under others (e.g., LED 4). Aggregating these patterns of all lights better differentiates pupil statuses.

3.2 Inferring Pupil Position and Size

With $M \times N$ data points (light intensity values) from the photodiodes at time t , the next step is to infer the 2D coordinate of the pupil center and pupil diameter at t . We solve the problem with supervised learning to train offline a personalized model capturing the relationship between the sensing data and pupil status. With the trained model, we then compute pupil position and diameter based on the current sensing data. Specifically, we choose boosted trees regression algorithm [22, 29] that optimizes a sequence of regression trees with weights associated to leaves (decisions). Each new tree helps in correcting errors made by the previously trained tree. The key benefit of boosted trees regression is its low complexity during real-time inference, which involves only comparison and addition operations. In our experiments, the time complexity of boosted tree regression is less than 10% and 25% of that using feed-forward neural networks and support vector machines (SVMs), respectively⁴. Also, the space complexity of boosted tree is less than 10% of that for SVMs, making it suitable for a low-power microcontroller. Random forest is another candidate because of its low computation overhead. However, boosted trees achieve exceptionally higher accuracy than random forest when dimensionality is low (e.g., < 4000) [20] and is calibrated [21], making boosted tree regression a better fit in our context.

Offline Training. We train a separate boosted trees model for pupil’s coordinate in x and y axis, and pupil diameter z , respectively. To train the models for a user, we collect

³In our context, an NIR LED with low radiant power and low duty cycle consumes power similar to that of a photodiode.

⁴As for tracking accuracy, we observe that boosted trees actually outperform more complex alternatives, with 3-mm and 0.2-mm smaller mean error than neural networks and SVMs, respectively. We hypothesize that it is because our scenario does not offer high-dimensional and massive volume of training data for these alternatives to excel.

5-minute data where we instruct the user to stare at a ball on the computer screen and follow its moving trajectory. The user wears our eye tracker that collects light-sensing data, as well as a wearable NIR camera that captures eye images (§6). We later manually label camera images to obtain the ground truth of pupil’s 2D positions and leverage the Pupil Labs [5] to acquire the ground truth of pupil diameter.

With $M \times N$ data points from light sensing at time t , we extract a feature vector \mathbb{F}_t with $(M \times N + M)$ dimensions, which contain the spatial variations of reflected light across photodiodes under each NIR light, as well as the mean reflected light intensity under each light. Specifically, let $s_{i,j,t}$ denote the reflected light intensity perceived by photodiode i under NIR light j at time t . We then compute the mean reflected light intensity across all photodiodes under light j as $\overline{s_{j,t}}$. We compose \mathbb{F}_t as:

$$\mathbb{F}_t = \{(s_{i,j,t} - \overline{s_{j,t}}), \overline{s_{j,t}} | 1 \leq i \leq N, 1 \leq j \leq M\}. \quad (1)$$

We use the feature vectors \mathbb{F}_t and pupil’s coordinate or diameter to train a tree ensemble of K trees with maximum depth h . We leverage five-fold cross-validation to fine-tune K and h and set K, h to 50 and 5 respectively, as they achieve the best tradeoff between computation complexity and accuracy.

Online Inference. As light sensing data arrive on the fly, we compute the feature vector as Eq. (1) and feed the feature vector to the trained tree ensembles to compute pupil’s coordinate in each axis and pupil size separately. Similar to the random forest, each tree will have a prediction score, and the pupil’s coordinate is the summation of the K scores.

To improve the energy efficiency of running online inference on a micro-controller (MCU), we eliminate most floating-point operations during online inference. Regression with boosted trees has a large number of floating-point comparisons and additions. Since all features are integers, we convert all comparisons to integers when deploying the regression model to the MCU. For additions, we approximate floating-point numbers by keeping three digits after the decimal point. By left shifting 10 bits, we first scale these floating-point number up to 1024 times and then store the results as integers. Some multiplication and division operations can be replaced by shift operations when the multipliers and dividers are multiples of two.

4 ADAPTIVE EYE TRACKING

The second design element further optimizes the energy efficiency of the eye tracking system at a macro level. It examines the sequence of inferred pupil positions to determine the current eye movement stage/type. It then exploits the movement characteristics during the current movement stage to adjust the sensing and computation on pupil inference. It further saves system energy without sacrificing tracking

accuracy. Next we first describe eye movement types and characteristics. We then introduce our adaptation strategy.

4.1 Eye Movement Patterns

Human eyes do not look at scenes with fixed steadiness. Eye movement falls into one of these four types/stages:

- **Fixation** is the stage when gaze focuses on a single location with little pupil movement (below 0.5°). Fixation lasts 150 ms to 500 ms [34, 52, 56, 61, 69].
- **Smooth pursuit** is the eye movement following a smooth trajectory with relatively constant velocity ($30^\circ/\text{s}$ on average), i.e., zero acceleration [17].
- **Saccade** is rapid eye movement in jumps with varying velocity, which can reach $700^\circ/\text{s}$ [14] with the minimum of $40^\circ/\text{s}$ [10]. A saccade lasts around 200 ms on average [27].
- **Blink** is the closing and opening of an eyelid to help spread tears and clean the surface of the cornea and conjunctiva [49]. A blink lasts 100 to 400 ms [2] and occurs 4 to 26 times per minute [13], depending on the activity (e.g., reading, conversation).

As examples, Figure 4 plots pupil’s 2D positions and the time series of velocity in various stages. They are samples from our dataset collected by a remote Tobii eye tracker [8].

Continuous eye movement is a sequence of transitions among these stages. As revealed by early research [33], when scanning a scene or reading, human eyes move in jerks and stop several times, resulting into a series of fixation points with saccadic movements or smooth pursuit in between. The same holds when people look at a picture [76] or watch a movie [30]. Figure 4(d) shows a snippet of eye movement sequence as an example, where the movement from location A to B contains two fixation stages, connected by saccadic and smooth pursuit movements.

This interesting eye movement pattern motivates us to quantify the percentage of each movement stage/type for a variety of common activities. We conduct a user study with 12 participants (10 males and 2 females, 18 to 50+ years old). We examine two types of scenarios: 1) *screen viewing*, where each participant sits in front of a laptop screen and performs various tasks including fast reading, slow reading, playing video games, and watching movies; 2) *conversation*, where participants are in a meeting and freely look around the environment or stare at other users. We collect participant’s eye movement data at 60 Hz using a remote Tobii eye tracker [8]. Our dataset contains half-an-hour data for each user in scenario 1) and 15 minutes for scenario 2). The frames of eye blinks are detected and marked by the Tobii eye tracker. We then classify remaining movements into different stages based on their movement velocity and acceleration [10]. Specifically, we apply a prior method [52], where movements with velocity below $5^\circ/\text{s}$ are marked as

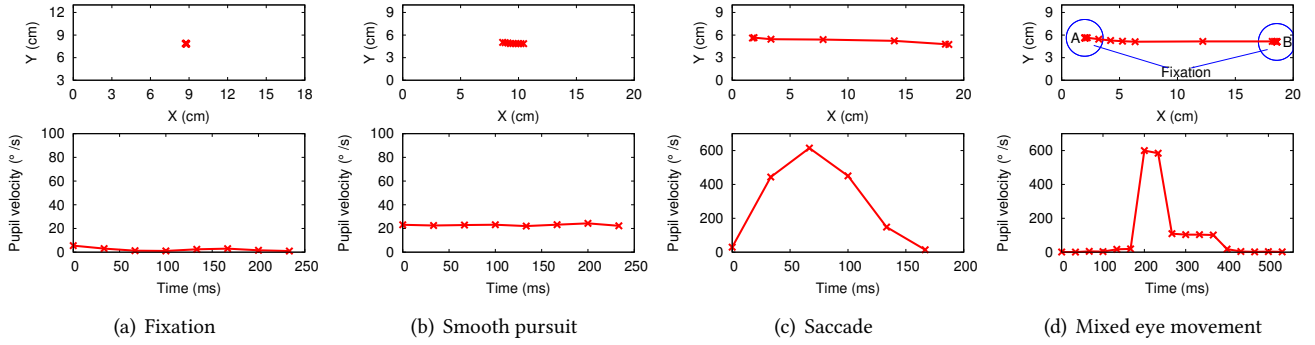


Figure 4: Pupil position (top) and velocity (bottom) during fixation, smooth pursuit, saccade, and mixed eye movement.

fixation, those with velocities from $5^\circ/\text{s}$ to $40^\circ/\text{s}$ and acceleration below $1^\circ/\text{s}^2$ are smooth pursuit, and those with velocities above $40^\circ/\text{s}$ are saccades. With fixed thresholds, this method can misclassify some movement stages; the results, however, can still indicate the high-level distribution of these stages.

Table 1 shows the percentage of each movement stage for each activity, averaged across all participants. We also include the standard deviation in parentheses. Our main observation is that regardless of the activity, the majority (80%+) of eye movements are in fixation or smooth pursuit, where fixation occupies a slightly larger portion (43 – 45%) than smooth pursuit (40 – 41%). The small standard deviation numbers indicate that the pattern is consistent across participants. One reason that a significant portion of our eye movements is fixations is that it takes time for eyes to fixate on the area of interest, and the brain acquires information during fixations [30]. The observation also aligns with prior studies on eye fixation patterns [16, 34]. In comparison, only 9 – 15% of eye movements are in saccades, even for tasks (e.g., fast reading) when users rapidly move their eyes all the time. During saccades, the information from the eye is mostly suppressed, and the eye will slow down its velocity to acquire information within areas of the scene [30]. Finally, less than 2% of eye movements are blinks.

	Fixation (%)	Smooth pursuit(%)	Saccade(%)	Blink(%)
Slow reading	48.32 (1.57)	40.54 (1.72)	9.71 (0.5)	1.43 (0.11)
Fast reading	45.25 (1.07)	41.52 (1.27)	11.34 (0.37)	1.89 (0.13)
Watching movies	43.84 (2.47)	41.31 (1.56)	14.07 (0.59)	0.78 (0.06)
Playing games	45.6 (4.22)	41.22 (3.97)	12.22 (1.3)	0.97 (0.09)
Conversation	60.85(5.35)	32.64(0.69)	5.26 (0.33)	1.25 (0.12)

Table 1: Average percentages of eye movement stages for five activities across 12 participants. We also include standard deviations in parentheses.

We further examine the predictability of eye movement in each stage (except blink). Using the dataset collected in the user study, we downsample the data at 30 Hz and apply linear interpolation to infer pupil positions in the skipped frames. Figure 5 compares errors of inferred pupil positions across movement stages. We observe that for fixation and

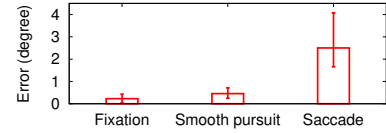


Figure 5: Errors of pupil positions inferred by linear interpolation in different stages of eye movement.

smooth pursuit, the mean errors are below 0.5° with the 95th percentiles below 1° , whereas for saccades, the mean error is 2.5° with 95th percentile above 4° . These results confirm the movement predictability within fixation and smooth pursuit. Saccadic movements in comparison are less predictable.

Overall, the fact that fixation and smooth pursuit dominate the majority of eye movements is encouraging. It indicates that exploiting the movement predictability during these stages can lead to a significant energy saving without sacrificing tracking accuracy. Next, we will describe our adaptation strategy in detail.

4.2 Adaptation based on Eye Movement

The characteristics of different eye movement stages are informative to the eye-tracking system’s operation. During eye fixation and smooth pursuit, prior pupil positions are highly predictive of the future. Thus, the system can predict the next sequence of pupil positions fairly accurately using *predictive inference*, without needing to switch on NIR lights and photodiodes to gather sensing data and run the full-fledged pupil inference algorithm. During an eye blink, sensing and pupil inference can be switched off as eyelids cover the pupil and sensing data are no longer relevant. Full-fledged sensing and pupil inference need to be performed at high rates only for saccadic eye movements.

The challenge of such an adaption is twofold. *First*, we must timely detect the entry and exit of a moving stage so that sensing and inference are adapted correctly. Quick detection of a movement stage also allows more future inferences within this stage to benefit from predictive inference for

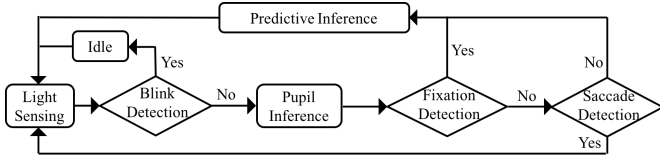


Figure 6: System flow of adaptive eye tracking.

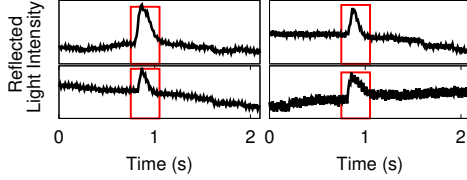


Figure 7: Reflected light intensity changes when the user blinks the eye. The red box represents a blinking instance

more energy savings. *Second*, since predictive inference relies on prior inference results, it inevitably introduces errors that can accumulate and propagate to future inferences.

We address these challenges as follows. *First*, we apply efficient mechanisms to detect each eye movement stage. We dynamically adjust our detection threshold based on the current noise level so that the detection is robust against noise-level changes. *Second*, once detecting the entry of a movement stage, we interleave predictive inference with the full-fledged sensing and pupil inference. We strategically adjust the frequency of running the latter at multiple levels. It helps periodically clear the accumulated error from predictive inference while ensuring quick detection of the transition to the next stage. Next, we begin with an overview of the system flow, followed by the detection and adaptation strategy for each eye movement stage.

System Flow. As shown in Figure 6, the system samples reflected light intensity values from photodiodes. Based on the sampled data, it first detects whether a blink occurs. If so, the system is switched off for the blink duration. Otherwise, it runs the full-fledged inference algorithm (§3.1) to infer pupil position. Based on pupil position trajectory, velocity, and acceleration, the system detects the eye movement stage and adjusts the rate of sensing and inference accordingly, with predictive inference to provide inferences for skipped frames. Full-fledged sensing and inference are performed at output frame rate only for saccadic movements. Algorithm 1 lists the details⁵ We next explain each step in detail.

Blink Detection & Adaptation. We exploit the spatial correlation of reflected light changes across photodiodes to detect blinks. Unlike pupil movement that weakens reflected light in some directions while strengthening it in others, a blink causes uniform changes in the reflected light across most photodiodes. Because eyelids reflect light when the

⁵We divide the sensing rate by two or four, because such divisions can be implemented as bit shifts, which run faster and consume less power.

Algorithm 1: Adapting the rate of sensing and inference.

input : 1) velocities in 40-ms window: V^x, V^y ; 2) acceleration: a_t ; 3) fixation thresholds: v_f^x, v_f^y ; 4) threshold factors for CFAR: $\alpha_{saccade}, \alpha_{blink}$; 5) reference samples and test samples: $R_{saccade}, R_{blink}$; 6) the number of LEDs and photodiodes: M, N ; 7) output frame rate r

output: Eye movement state st , the rate of full-fledged inference r'

```

blink_counter = 0;
for i ← 1 to M do
    for j ← 1 to N do
        if  $s_t^{ij} - \min(R_{blink}^{ij}) >$ 
             $\alpha_{blink} \cdot (\text{median}(R_{blink}^{ij}) - \min(R_{blink}^{ij}))$  then
            | blink_counter ++;
        end
    end
end
if blink_counter >  $M \times N / 2$  then
    st = blink;
    r' = 0; switch off the system for 200 ms;
end
else if  $V^x < v_f^x$  &  $V^y < v_f^y$  then
    st = fixation;
    r' = r/4;
     $T_f = \text{duration}()$ ; // fixation duration (ms)
    if  $T_f > 200$  then
        | r' = r/2;
    end
end
else if  $a_t > \alpha_{saccade} \cdot \text{median}(R_{saccade})$  then
    | st = saccade; r' = r;
end
else
    | st = smooth pursuit; r' = r/2;
end
end

```

eyelid covers the pupil, it results into stronger reflected light perceived by photodiodes; when eyelid opens, pupil continues absorbing light and causes a significant drop in reflected light intensity for most photodiodes. As a result, eyelid during a blink create a pulse in the time series of reflected light intensity perceived by most photodiodes. Figure 7 illustrates an example for four photodiodes when a NIR light is on, where each line represents the time series of perceived light intensity at a photodiode.

Based on this phenomenon, blink detection boils down to detecting a dramatic change in reflected light across most photodiodes. A simple method is to compare photodiode signals to a fixed threshold and examine if most photodiodes perceive signal jumps. To set a proper threshold, however, is difficult, as it is subject to the current noise level that can vary both spatially and temporally. We address this problem by applying the constant false alarm rate detection (CFAR)

to estimate an adaptive threshold on the fly. CFAR is widely used in the radar systems to detect dramatic signal changes in the noisy background [66] and has been applied in the eye tracking context [11, 51]. In a nutshell, CFAR estimates current noise level by examining m reference samples around the current test sample (i.e., current sensing data in our context). It excludes n samples (i.e., guard samples) adjacent to the test sample to avoid corrupting the noise estimate with the test sample. By adapting the threshold, CFAR maintains a constant probability of false alarm.

We slightly adjust CFAR in our context, as traditional CFAR considers reference samples with the test sample in the center [66], whereas here test sample is the latest sensing data. Thus, we consider m samples before the test sample as the reference samples. Let s_t^{ij} denote the test sample, i.e., the light intensity value at photodiode i when NIR light j is on. The set R_{blink}^{ij} of reference samples for this pair of photodiode and light contains $s_{t-n-m}^{ij}, s_{t-n-m+1}^{ij}, \dots, s_{t-n-1}^{ij}$. Then a blink is detected if the following condition holds for at least half of the light-photodiode pairs:

$$(s_t^{ij} - \min(R_{blink}^{ij})) > \alpha \cdot (\text{median}(R_{blink}^{ij}) - \min(R_{blink}^{ij})), \quad (2)$$

where α is the threshold factor. It is calculated as [51]:

$$\alpha = f(1 - P_{fa}) / f(1/2), \quad (3)$$

where f is the error function⁶, and P_{fa} is the false alarm rate. In our implementation, we set m , n , and P_{fa} as 20, 10, and $1e-2$ respectively.

Once detecting a blink, the system switches to the idle mode for 200 ms without performing any sensing or inference, given that a blink lasts around 250 ms on average [2]. After 200 ms, the system continues sensing and full-fledged inferencing at its output frame rate r . Based on the inferred pupil positions, we next introduce the detection of different pupil movement stages and corresponding adaptation.

Fixation Detection & Adaptation. Inspired by prior studies [34, 56, 61], we detect fixation using a threshold-based method. Since the mean fixation duration is around 250 ms, we examine pupil positions within a 40-ms sliding window (or 6 frames @120Hz) to determine the entry of fixation. Let (x_t, y_t) denote the pupil's position at time t , and T be the time interval between two adjacent inferences. We estimate pupil's movement velocity in x - and y - axis at time t as:

$$v_t^x = \frac{x_t - x_{t-2}}{2T}, \quad v_t^y = \frac{y_t - y_{t-2}}{2T}. \quad (4)$$

If both velocities are below threshold v_f^x and v_f^y respectively, then the fixation test at time t is passed. If such test is passed for all positions in the sliding window (i.e., from time t to $(t-5)$ @ 120Hz), then the system marks the current stage as

⁶ f is a Gaussian error function in our implementation as we observe that sensor errors follow a zero-mean normal distribution.

fixation. We set v_f^x as $5^\circ/\text{s}$ and v_f^y as $3^\circ/\text{s}$, based on prior studies [38, 67, 69] and our experiments⁷. The system keeps conducting this fixation test for every subsequent pupil position. It exits the fixation stage whenever the test fails, ensuring that the system can timely detect the exit of fixation.

Upon the detection of the entry to fixation, the system lowers the frequency of conducting sensing and full-fledged inference to $r/4$ for the next 200 ms, where r is the output frame rate of inference results. The skipped frames in between are provided by predictive inference to maintain the output frame rate as r . Predictive inference for fixation is simply the previous pupil position given that pupil rarely moves. Within the 200 ms, if the fixation test fails at any time point, the system immediately treats it as the exit of fixation and returns the frequency of sensing and inference to r . If the fixation stage remains after 200 ms, statistically pupil is likely to exit fixation anytime soon. To ensure timely detection of exit, the system increases the frequency of sensing and full-fledged inference to $r/2$.

Upon the detection of the exit of fixation, the system further examines whether the current movement is saccadic using the method we next describe.

Saccade Detection & Adaptation. Saccade detection is based on sudden, significant jump in acceleration. We use acceleration rather than velocity because the velocity ranges of smooth pursuits and saccades overlap [11]. Instead of setting a fixed threshold for the acceleration, we also apply the CFAR method to estimate the proper threshold on the fly and detect the entry of a saccade. In comparison to a fixed threshold, the adaptive threshold is more robust against noises that can change both spatially and temporally. If the threshold is too high, many saccades will be treated as smooth pursuits, which degrades tracking accuracy. If the threshold is too low, the system will miss many smooth pursuits and thus opportunities for energy saving. CFAR adapts the threshold to maintain a constant probability of false alarm, which balances the tracking performance and energy efficiency.

Specifically, let a_t denote current acceleration (i.e., test sample) at one axis. We derive the adaptive acceleration threshold for each axis by estimating the noise level around a_t . Similarly to blink detection, the set $R_{saccade}$ of m reference samples contains $a_{t-n-m}, a_{t-n-m+1}, \dots, a_{t-n-1}$. Then the entry of a saccade is detected if the following condition holds in both x - and y - axis:

$$a_t > \alpha \cdot \text{median}(R_{saccade}), \quad (5)$$

where α is the threshold factor calculated as Eq. (3). We set m , n , and P_{fa} as 20, 10, and $1e-3$ respectively in the implementation. To minimize the computation overhead, we

⁷Adaptive thresholds bring negligible gains for fixation detection because velocity is near zero.

rectify a_t by an absolute value operator. With this simple design, online saccade detection mainly involves computing the median of m numbers in $R_{saccade}$. We leverage a red-black tree and a circular buffer to minimize the overhead.

Once detecting a saccade, the system maintains the frequency of sensing and full-fledged inference as r . Otherwise, the system marks the current stage as a smooth pursuit, sets the full-fledged inference rate as $r/2$ and applies linear interpolation as the predictive inference to infer skipped frames.

5 PROTOTYPE IMPLEMENTATION

We have built a compact and lightweight (< 25 g) prototype using off-the-shelf hardware components. The prototype comprises three units (Figure 8).

Light-Sensing Unit. We design and fabricate a thin (0.8-mm) PCB (< 10 g) that hosts 6 NIR LEDs (VSMY2943) and 12 photodiodes (BPV22F), both with peak spectrum sensitivity at 940 nm. We slightly tilt LEDs and photodiodes so that each LED’s 3dB-beam covers 70% of eye surface and each photodiode perceives 50% of eye surface within its 3dB field-of-view. We choose BPV22F for two reasons. *First*, its spectral sensitivity range is narrow (80 nm) and far from the visible light spectrum, which filters out ambient visible light interference. *Second*, with $\pm 60^\circ$ 3dB field-of-view, it can perceive more reflected light rays from the eye even at a short distance (< 2 cm). We choose VSMY2943 because it leverages Vishay’s new surface emitting technology, which saves up to five times energy compared with standard emitter technology [9]. We add a current flow control component to trigger LEDs sequentially and limit the radiant intensity to 0.1 mW/sr, which is much lower than the infrared irradiance standard for eye safety [28]. We add three 4-channel, low-power amplifiers (OP481) to amplify signals from photodiodes. Each amplifier contains a built-in low-power supply (4 μ A per amplifier at maximum), low-offset voltage (1.5 mV), and low-bias current (3 nA), which help further reduce the power consumption of the sensing unit and improve the signal-to-noise (SNR) ratio.

Computing Unit. We use a MINI-M4 FOR MSP432 board⁸ to digitize analog signals from amplifiers, extract features, detect eye movement stages, and infer pupil positions. MSP432 is an ultra-low-power MCU with 80 uA/MHz in the active mode. It embeds a low-power ADC with 400 uA at 1 Msps. During signal digitization, we leverage the embedded Direct memory access (DMA) in MPS432 to maximize ADC rate and reduce energy consumption. Inference results are stored in the MCU. They can also be transmitted to other devices through a UART port or battery-free wireless transmissions, which we leave to future work.

⁸We have removed all irrelevant components (e.g., USB bridge and LED indicators) on the MINI-M4 FOR MSP432 board.

Energy Harvester. The system harvests energy from ambient light to power all its operations. We choose light because of its higher energy density compared to other energy sources [73]. The harvester leverages two 11.4×3.7 cm Thin-film solar cells (LL3-37), which provide high short-circuit current with improved harvesting efficiency [71]. Also, as thin (0.2 mm), bendable films, they are lightweight and flexible to attach to the side arms of glasses (Figure 1).

The harvested energy can vary due to user movement (e.g., walking, head movement) and ambient light fluctuations. It can even occasionally fall below system’s required power. To deal with energy fluctuations, we use a buck-boost DC/DC converter (LTC3016) and a super-capacitor (0.22 F). LTC3016 stabilizes the output voltage (at 3.3V), while the super-capacitor stores extra energy when the harvested energy exceeds the currently consumed power. The stored energy can supply the system when the instantaneously harvested power is insufficient.

6 PROTOTYPE EXPERIMENTS

We recruit 22 participants (8 females) to evaluate our prototype⁹. Two of them wear contact lens. Table 2 summarizes participant information.

	Eye Color			Skin Color		
	Black	Blue	Green	White	Yellow	Black
# of Users	16	5	1	6	11	5

Table 2: Participant information.

Experimental Setup. We augment our prototype with a small infrared camera (PI NoIR [4]) to acquire the ground truth. Specifically, we install the camera in front of the eye (2.5 cm away) using a 3D-printed holder glued to our prototype (Figure 9)¹⁰. We connect the camera to a Raspberry Pi board and set the frame rate to its maximum (120 FPS) and image resolution to 320×240 . To synchronize camera images and light sensing data, we add three NIR LEDs (940 nm) next to the camera and program these LEDs to emit an 8-ms light pulse at the beginning of each experiment. It leads to a sudden brightening of the image and a rise in photodiode readings. We exploit this feature to identify the first camera image frame and the first sample of light sensing data. We conduct a separate experiment with 12 out of 22 participants to evaluate the pupil diameter inference. In the experiment, the camera runs at 60 FPS with image resolution of 640×480 .

With this augmented prototype, we collect training and testing data from each participant. The training phase lasts 5 minutes, where each participant wears our prototype, sits in front of a computer screen, and stares at a moving red ball on the screen. The red ball is programmed to move along

⁹We obtained IRB approval for our study from the local institution.

¹⁰In the future, we will consider using remote cameras (e.g., front cameras of smartphones or laptops) to ease the collection of ground truth.

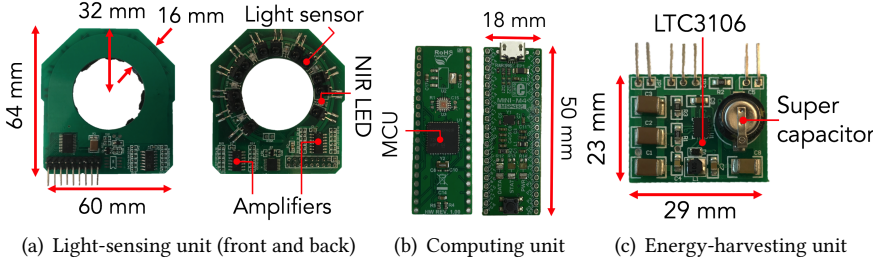


Figure 8: Circuit boards of each unit in our prototype.

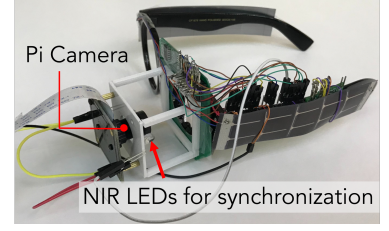


Figure 9: Augmenting our prototype with a NIR camera capturing the ground truth.

a smooth zigzag path scanning the screen. This maximizes the number of pupil positions in our training data during this short data-collection period. Overall we have collected roughly 35,000 frames on average per participant (769,710 frames for all participants). To obtain the ground truth, we manually label the pupil center and blink status in each image frame. We leverage the Pupil Labs [5] API to extract the pupil shape (e.g., oval) and derive the pupil diameter as the length of the oval’s long axis. To improve labeling accuracy, we skip blurred or overexposed images. We use light sensing data and the ground truth to train a personalized inference model for each participant. The offline training is done on a desktop (iMac 2012) and the computation takes less than 10 seconds.

In the testing phase, each participant performs the five activities in Table 1 (§4.1) while wearing our prototype running the trained personalized model. Our system computes and stores inferred pupil positions and diameters on the fly. Each activity lasts one minute, during which participants can freely move their heads and choose the content or scene to view. In total, we have collected 721,846 frames for testing. Similarly, we obtain the ground truth through manual labeling. We then compare them to our online inference results to evaluate the prototype’s tracking performance.

Error Metric. Similar to a recent work [45], our error metric is the distance deviation between inferred and ground-truth pupil center or diameter. We do not choose angular error as the metric because deriving the gaze vector from an image accurately (e.g., $< 0.5^\circ$ error) requires either multiple cameras to extract optical axis of the eye, or high-resolution images (e.g., 640×480) to locate the glint position on the image. Our hardware setup has only one wearable camera with image resolution of 320×240 at 120 FPS (the output rate of our inference). Thus, estimated gaze vectors are less accurate than labeled pupil centers and diameters.

6.1 Accuracy & Precision

Overall Accuracy. We consider two variants of our method for comparison: 1) running sensing and inference at 120 Hz without the adaption in §4, and 2) applying adaption with fixed thresholds for detecting eye movement stages. The first

variant serves as an upper bound to examine possible sacrifice in accuracy by adding predictive inference. The second variant examines the contribution of adaptive thresholds.

Figure 10(a) plots CDFs of tracking errors of pupil positions for all participants. Overall, our system achieves 0.8-mm mean tracking accuracy with 2.3 mm at the 95th-percentile and 4 mm as the maximum. Comparing it to the variant without any adaptation, we observe the negligible difference (< 0.01 mm). It indicates that our adaption mechanism properly skips sensing and inference during predictive eye movements and thus entails minimal sacrifice on tracking accuracy. Comparing to the variant with fixed thresholds, our method is most effective in improving the tail performance, with a reduction of 0.77 mm at the maximal error and 0.4 mm at the 95th percentile. The reduction is moderate because adaptive thresholds are only applied for detecting blinks, saccades, and thus smooth pursuit. The reduction in tracking error for these types of eye movements, however, is valuable, since measures of these movements are effective clinical metrics [23, 35]. Our results show adaptive thresholds are effective in identifying eye movement transitions.

Figure 10(b) plots the CDF of tracking errors in pupil diameter. The mean error is 0.3 mm with 0.9 mm at the 95th-percentile and 3.4 mm as the maximum. We observe slightly larger errors during saccades. The reason is that reflected light is affected by both pupil size variation and pupil movement. During saccades, the impact of pupil movement on reflected light is more significant than that of pupil size variation, which degrades the accuracy of size inference. Currently, we train a separate model for inferring pupil position and size. In the future, we will consider a joint model [55] to combine pupil position and size in the training phase.

Error Spatial Distribution. We analyze the distribution of tracking errors in the spatial domain and plot the average tracking error for each pupil center position in Figure 10(c). We observe that for 80%+ positions, the mean tracking error is less than 1 mm. The larger errors mostly occur when the pupil moves to the vision boundary. In these scenarios, the pupil is partially occluded, which degrades the accuracy.

User Activity. Our eye movement patterns can vary across activities, which in turn affect the manner our system adapts

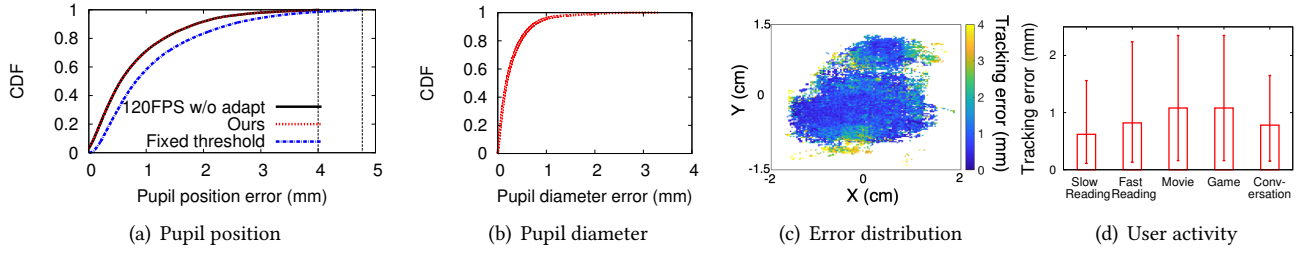


Figure 10: Accuracy of tracking pupil position (a) and diameter (b) based on the results of all participants. (c) plots the spatial distribution of pupil position errors and (d) shows the error across activities.

its sensing and inference. To gain a better understanding of the impact of user activity, we analyze tracking accuracy across different activities. For each activity, we aggregate the results across participants and plot the average tracking error in Figure 10(d). We also include error bars covering 90% confidence intervals. We observe that tracking errors during gaming and watching movies are slightly higher than others. Further analysis shows that the percentages of saccadic eye movements are much higher during these activities (13% and 12% respectively). As a result, the mean velocity and acceleration of pupil movement are at least three times higher than that of other activities. With 120-Hz tracking rate, it takes at least 8.3 ms for the system to detect transitions of pupil movement stages, which introduces errors in capturing rapid transitions. In §8, we will discuss our plan to increase the tracking rate without raising the energy budget.

User Diversity. We further analyze how tracking accuracy is affected by individual differences in skin/eye color and eye movement pattern. We aggregate each participant’s inference results (22 users for the inference in pupil position and 12 users for the inference in pupil size) and plot in Figure 11 the average and 90% confidence interval as the error bar. Our observations are as follows. First, for all participants, the mean error of the inferred pupil position and diameter is below 1.1 mm and 0.4 mm, respectively. Furthermore, as we compare mean errors across participants, we observe that the differences of mean errors in tracking pupil position and diameter are within 0.8 mm and 0.15 mm, respectively. In particular, as for tracking pupil position, the results of user 1 are the most accurate (0.5-mm mean, 1 mm at the 95th percentile), while that of user 4 have the largest error (1.3-mm mean, 3.2 mm at the 95th percentile). Analysis of user 4’s data reveals that this participant has the highest mean pupil acceleration and velocity during the experiment, leading to higher tracking errors. The results overall show that the system is more robust against individual differences than a prior work [41], which tracks pupil position using visible light. The main reason is the use of near-infrared light as the sensing medium, whose reflection is less affected by the skin or eye color than that of visible light.

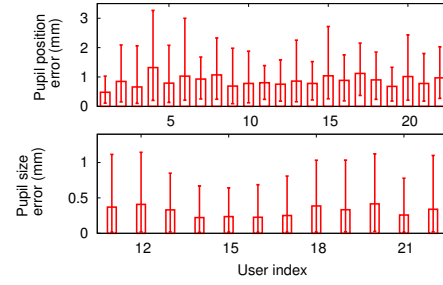


Figure 11: Mean errors in tracking pupil position and diameter across participants. The error bars indicate 90% confidence intervals.

Precision. We also examine the system’s precision, a measure of stability and reliability of pupil position inferences under a fixed gaze. It reflects the ability of the eye tracker to reliably reproduce a measurement and has been widely adopted by existing eye-tracking research [31]. Precision is computed by the root mean square (RMS) from the successive data points when the user looks at a fixed point. The mean precision for our system is 0.4 mm and the large errors mostly occur around the vision boundary.

6.2 Detection of Eye Movement Types

Next, we evaluate the accuracy in detecting various eye movement stages (blinks, fixation, and saccade). In Table 3, we list the precision, recall, and F1 score based on the data of all participants. Precision is the ratio of correctly detected blink instances within all detected blink instances. Recall is the ratio of detected blink instances within all actual blink instances. The F1 score is the harmonic average of precision and recall. Overall, all movement stages are detected with 90%+ accuracy. For blink detection, some false detections are due to the sudden movement of the glasses frame. It causes drastic signal changes at most photodiodes and can trigger the system to treat it as a blink mistakenly. For fixation detection, its errors are related to tracking errors of pupil position, since a fixation is detected based on the pupil movement velocity ($3^\circ/\text{s}$ or 1.5 mm/s). Large errors in pupil position (e.g., $> 1.5 \text{ mm}$, 15% of the tracking results in Figure 10(a)) can cause wrong fixation detection. Additionally, for all tested

activities (Table 1) with all participants, we observe blinks, fixations, and saccades occupy 1.19%, 49.37%, and 8.16% of the total frames, leaving 41.27% for smooth pursuits. The result aligns with prior observations in Table 1. It confirms that regardless of user activities, eye movement is predictable in the majority of cases because of its jerky nature.

	# of Frames	Precision	Recall	F1
Blink	8,589 (1.19%)	0.933	0.947	0.94
Fixation	356,375 (49.37%)	0.921	0.916	0.92
Saccade	58,902 (8.16%)	0.918	0.935	0.93

Table 3: Accuracy of detecting eye movement types.

6.3 Latency

We next examine tracking latency, which is defined as the duration to produce a pupil inference. Table 4 lists the latency of each key step to produce an inference. Overall, pupil inference and the movement detection run fairly fast, and the light sensing step dominates the latency. The light sensing step includes sequentially switching on each NIR LED while photodiodes sense reflected light under each LED. The latency comes from three components: 1) 40 μ s for the pinout setups on the micro-controller; 2) 360 μ s for acquiring a set of $M \times N (= 72)$ data points, given that the micro-controller samples at 200 Ksps; and 3) 10- μ s delay before switching to another LED, which helps photodiodes and amplifiers better respond to NIR light intensity change. The inference computation takes 160 μ s to infer a pupil position. For higher efficiency, inference subtasks are executed during each 10- μ s switching delay. It results in an overall latency of 560 μ s for one inference, potentially supporting tracking above 1.7 KHz.

Steps	Light Sensing	Movement Detection	Pupil Inference	Total
Latency (μs)	450 (± 20)	<1	160 (± 5)	560 (± 25)

Table 4: Tracking latency and its breakdown.

6.4 Energy Consumption

We now examine the power consumption of our prototype. We measure the power consumption using a Monsoon power monitor. Table 5 lists the energy consumed by the key components to produce an inference result. Pupil inference here refers to the full-fledged inference in §3.1. Overall, each inference consumes less than 6 μ J, with signal digitization (2.5 μ J) and the inference algorithm (2.4 μ J) as the main contributors. Light emission and sensing consume less than 1 μ J (< 17% of the total consumption), where the peak power of each NIR LED is less than 0.5 mW. We can possibly further reduce the energy consumption with ultra-low power ADC (e.g., ADS7042). We leave it to future work. When the system does not perform any sensing or computation (e.g., during a blink), the MCU remains at the ultra-low-power mode and the light-sensing board will be shut down. The power consumption in this mode is less than 40 μ W.

	Light-Sensing Unit		Micro-Controller		
	Photodiodes	IR emitters	ADC	Movement Detection	Pupil Inference
Energy (μJ)	0.5 (± 0.04)	0.4 (± 0.05)	2.5 (± 0.08)	<0.1	2.4 (± 0.07)

Table 5: Energy consumption for producing one pupil inference.

We then further examine the power consumption with the adaption strategy in §4. Since the actual adaptation depends on user activities, we plot the average power consumption for each activity in Figure 12(a). We also include the result for the method without any adaption (i.e., running full-fledged sensing and inference at 120 Hz) as a baseline. The error bars cover 90% confidence intervals. Overall, the mean power consumption of our system across all activities is 395 μ W, which is 52% of the power consumed by the baseline without any adaption. The energy saving comes from the predictive eye movement types (i.e., fixation, smooth pursuit) and blinks. As shown in Table 1 and §6.2, these predictive movement types occupy 86% of all frames even in activities with rapid eye movement (e.g., gaming). This is because of the inherent characteristics of our eye movement (moving in jerks). Thus, a large portion of inferences is realized as predictive inferences using our adaptation scheme, leading to significant energy savings. Across activities, the power consumption under games and videos is slightly higher. It is because more saccades occur during these activities and the system skips fewer frames to run the full-fledged sensing and inference.

Next, we compare our system with four existing low-power eye trackers (LiGaze [41], CIDER [45], iShadow [44], and iGaze [77]), by examining their tradeoff between tracking accuracy and power consumption. Since each system runs at a different tracking rate, we compare them by energy consumption per inference. Given that some [41, 44, 77] report only angular errors, we convert these angular errors to distance deviations for a fair comparison. In particular, we leverage an emmetropic human adult eye [12], where the normal horizontal rotation range of an eye is $[-35^\circ, 35^\circ]$, and the pupil horizontal movement is within $[-18 \text{ mm}, 18 \text{ mm}]$. Thus, 1° angular error maps to $\frac{18}{35}$ mm distance deviation. Figure 12(b) shows the comparison. We observe that the power consumption of our system is several orders of magnitudes lower than existing methods without sacrificing much in tracking accuracy. Our system’s tracking performance is comparable to CIDER, which achieves the best tracking performance using cameras.

6.5 Energy Harvesting

Finally, we evaluate the energy-harvesting unit in typical room settings and show its capability to power the entire system indoors. Although solar energy harvesting has been studied extensively in the literature, to the best of our knowledge, there have been no systematic measurements with setups similar to ours (solar cells vertically placed on the

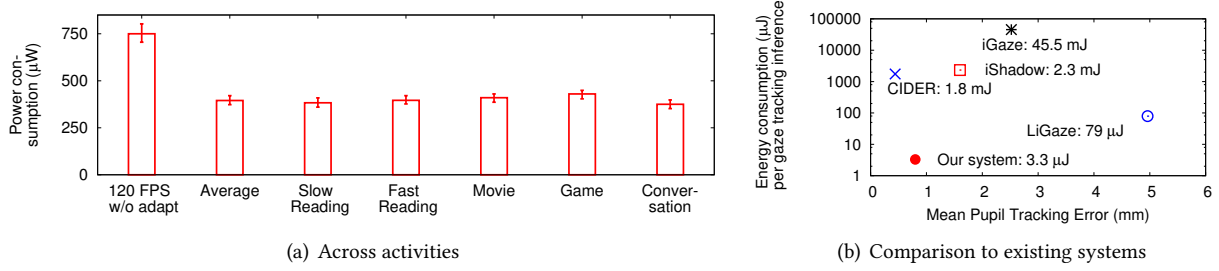


Figure 12: Power consumption across activities with and without the adaption in §4. We also compare it to existing low-power eye trackers.

glasses side arms) under various user activities. We test four ambient light settings, with light intensity ranging from 300 lux to 900 lux: 1) a 60 m² office with four fluorescent lights on the ceiling (2.6-m height). The light intensity at day and night is 700 lux and 500 lux, respectively, at a 70-cm high table; 2) the same office as 1), where we add three LED floor lamps emitting lights from sides; 3) the same office as 1), where we only turn on the floor lamps. The light intensity at day and night is 500 lux and 300 lux, respectively, at a 70-cm high table; 4) a 18 m² lab space with LED panels on the ceiling. The light intensity at day and night are 900 lux and 800 lux, respectively, at a 70-cm high table. We choose these settings based on the recommended light levels by the US national research & development center [6] and Illuminating Engineering Society (IES) [3]. The illumination for normal office work, library, laboratories is recommended to be above 500 lux. The 300-lux setting is below the standard and is used only to test low-light conditions. In each setting, we test scenarios when the user is sitting, standing, or walking while wearing our prototype with solar cells on the glasses arms.

Figure 13 plots the harvested power in each setting. We have three observations. *First*, both light source positions and user activities affect the amount of energy we can harvest. Since most indoor lights are on the ceiling, when users stand up or walk, solar cells are closer to light sources and harvest more energy. For floor lamps, some of their emitted light rays are perpendicular to the solar cells on glasses arms, allowing more power to be harvested. *Second*, across the luminary types, our energy harvester acquires more power under fluorescent lights than LEDs. As a less efficient luminary type, fluorescent lights radiate more heat and contain more infrared light, which can be converted to energy more efficiently by solar cells. *Third*, the harvested power is above the system’s requirement (395 μW) in most settings when ambient light is above 500 lux, except at night when the user is sitting on a chair. Under the low-light condition (e.g., 300 lux), the super-capacitor, which stores surplus energy from other light conditions, can compensate for the power gap. For example, after one-hour normal usage in setting 2 during the daytime, the surplus power in the super-capacitor can support the system for one hour in setting 3 at night.

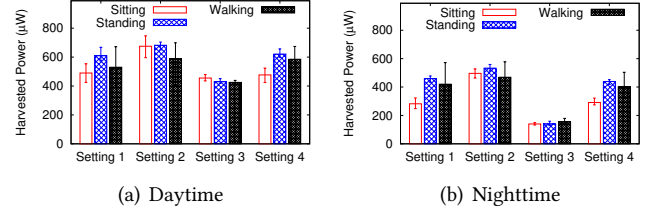


Figure 13: Harvested power from indoor lighting in four settings while a user wearing our prototype is sitting, standing, or walking.

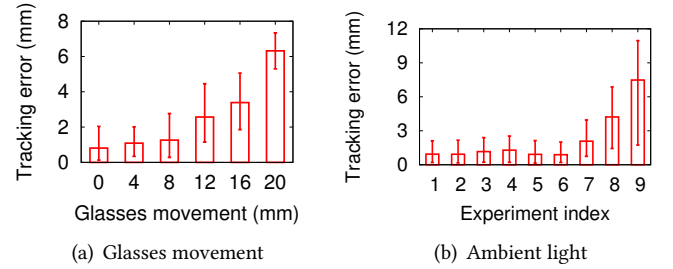


Figure 14: Impact of practical factors on tracking performance.

6.6 Practical Considerations

Glasses Movement. Glasses can slightly move during reading or user movement (e.g., walking). To analyze its impact on system performance, we instruct a participant to wear our prototype and to slightly move the glasses frame from the normal position on the nose by various offsets up to 2 cm, where with 2-cm offset, the glasses frame is on the tip of the participant’s nose. Figure 14(a) shows the mean tracking error of the pupil position, where error bars cover 90% confidence intervals. We see that the error increase is very small when the offset is within 8 mm. In these cases, the eye center changes up to 2 mm in the camera’s view, and the sensing data changes up to 2%, compared with that without any offset. Thus, the regression model can handle these small deviations in features/labels. However, for larger offsets (e.g., 2 cm), the eye center can move up to 15 mm in the camera view and sensing data can change up to 20%, leading to much larger errors. To enhance system’s robustness against glasses movement, we can collect more training data with various glasses offsets. We leave it to future work.

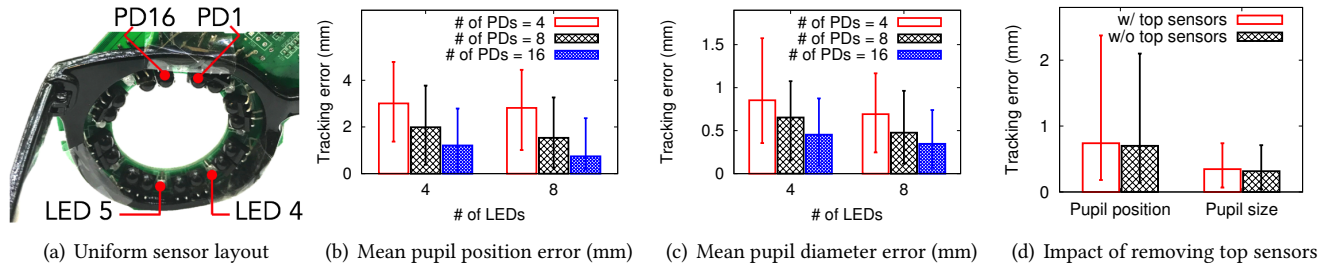


Figure 15: The impact of the number and placement of LEDs and photodiodes on eye-tracking performance.

Ambient Light. We also test system performance under nine ambient light conditions with various types of light sources, light directions, and light intensity levels. Table 6 shows the mean and standard deviation of sensor readings¹¹ when a user wears our prototype with all NIR LEDs switched off. Thus, these data indicate ambient NIR energy levels in various indoor settings. Figure 14(b) shows the tracking error of the pupil position, including 90% confidence intervals, in each light condition. Overall, our system is robust across these conditions, except strong direct sunlight (10^4 lux), which contains strong NIR energy. NIR energy radiated by indoor light sources or indirect sunlight has a negligible impact on our system. In these settings, the ambient NIR energy maps to sensor readings within 100 to 150, whereas with our NIR LEDs, sensor readings vary between 1500 and 11000 in the dark environment. However, direct sunlight maps to sensor readings up to 2500, which can saturate sensors when our NIR LEDs are on and degrade the tracking performance. To mitigate this problem, we can adapt the sensor gain to avoid saturation. We leave it to future work.

#	Type of Light Source	Light Source Position	Ambient Light Intensity (Lux)	Sensing Data (Mean/Std.)
1	LED Light	Uniform	300	5 / 1
2	LED Light	Uniform, Front Back, Left, Right	600	7 / 2
3	Fluorescent Light	Uniform	600	30 / 7
4	Incandescent Light	Uniform	600	47 / 11
5	LED Light	Uniform	800	35 / 9
6	Non-direct Sunlight	Window	1000	85 / 16
7	Non-direct Sunlight	Window	2000	141 / 26
8	Direct Sunlight	Window	5000	934 / 86
9	Direct Sunlight	Window	10000	2526 / 205

Table 6: Impact of ambient light on the NIR sensors of our prototype.

Sensor Layout. The number and placement of NIR LEDs and photodiodes are important design choices for the system. To examine their impact, we fabricate a prototype (Figure 15(a)) with 16 photodiodes (BPV22F) and 8 NIR LEDs (VSMY2943), which are the maximal numbers of photodiodes and LEDs physically allowed by the prototype’s form factor, considering all the associated circuits. We then sample sensing data from a subset of photodiodes and/or LEDs and

evaluate the resulting tracking performance. Figure 15(b) and 15(c) plot the mean errors in tracking pupil position and diameter under various configurations of photodiodes and LEDs. We also include error bars covering 90% confidence intervals. As expected, tracking errors gradually increase with fewer photodiodes and LEDs because of the decrease in sensing data. In particular, the number of photodiodes has more significant impact on the performance than the number of LEDs. It motivates us to keep the maximal number of photodiodes and LEDs in the system. However, as we further examine the sensing data from individual LED-photodiode pairs, we observe that because of the blockage of eyelashes, sensing data collected by top photodiodes or under top LEDs are typically noisier than others. We then remove the data from top 4 photodiodes and top 2 LEDs, and compare the resulting tracking performance to that without their removal. As shown in Figure 15(d), the removal of these top sensors does not hurt the tracking performance at all; rather, it leads to slightly lower tracking errors. Thus, we settle on the final design with 6 LEDs and 12 photodiodes with the placement illustrated in Figure 3(a).

7 RELATED WORK

Eye Tracking Systems. We divide existing eye-tracking systems into three categories based on their methodologies:

1) *Camera-based:* Most existing eye trackers use cameras. To achieve robust tracking indoors, existing methods rely on NIR LED to illuminate eyes and highlight the pupil [26, 48, 50, 54, 79]. Prior works have also examined the use of camera images under visible light [32, 39, 40, 57]. With the rise of wearable devices, several efforts have examined eye tracking in the wearable context [44, 45, 77]. Specifically, iGaze [77] leverages eye tracking to infer user attention on web browsing. The system achieves 5° tracking accuracy and 30-Hz tracking rate, but consumes over 1.5-W power. iShadow [44] and CIDER [45] sample image pixels to reduce the camera’s power consumption and computational overhead. The latest design [45] achieves 0.44-mm mean accuracy with 7-mW power at 4 Hz. The energy consumption is still a few orders of magnitudes higher than the energy that can be harvested

¹¹ 12400 is the maximal reading with 2.5-V reference voltage.

from ambient environment. Our method further reduces the energy consumption by replacing cameras with photodiodes and adapting eye tracking based on eye movement types.

2) *Photodiode-based*: Prior works have also examined the use of low-cost photodiodes for eye tracking. In [36, 59, 75], authors leverage a flying-spot laser and a photodiode to selectively image landmarks on the eye for eye tracking. In [65], Sarkar *et. al* design a customized MEMS mirror that guides laser rays to scan and track the cornea. Their method provides high accuracy ($<0.4^\circ$) and high-rate tracking (>1 kHz), using sophisticated optical lenses to control laser rays. In [68], authors embed a customized photodiode array into an LCD display for eye tracking. In comparison, our method does not require specialized hardware and uses off-the-shelf photodiodes/LEDs. A recent work [41] designs a low-power eye tracker in VR using low-cost photodiodes to sense screen light. The design relies on the modeling of the relationship between incoming screen light and reflected light. It is applicable to VR with a closed space where the screen is the sole light source at a fixed direction/distance. Generalizing it to regular glasses or augmented glasses in open spaces is difficult, given the uncontrolled and diverse nature of surrounding ambient light sources. Training the model entails heavy overhead and introduces tracking errors. Our work addresses the problems using multiple NIR light sources. It improves tracking accuracy and robustness. More importantly, our work proposes an adaptation strategy to further reduce power by exploiting eye movement patterns.

3) *Non-Vision*: Other eye tracking systems rely on the magnetic field or Electro-oculogram (EOG) sensors. In [60], David *et. al* use a contact lens with a scleral coil to track pupil with 1° tracking accuracy. In [18, 72], Bulling *et. al* use EOG sensors on wearable glasses to detect eye movement. Our method differs in that our system is less intrusive and consumes significantly lower power.

Eye Movement Analysis. Existing eye movement detection relies on dispersion, velocity, and acceleration thresholds [25, 64]. Suppes *et. al* present one dispersion-threshold algorithm to identify fixations as groups of consecutive points within a particular dispersion or limited region [70]. Smeets and Hooge [67] present a multipass velocity-based saccade detection in which they detect saccade peaks in the first pass and find the onset/offset in the second pass. In [11, 51, 52], adaptive thresholds are used for saccade detection to deal with differences in users, recording situations, and noises. Extensive works also have analyzed user behaviors using eye movement detection. In [24, 30, 34], the authors extract user's reading patterns of texts, video, and web search using fixation duration and saccade frequency. In [23], Trevor *et. al* leverage saccade behaviors to detect early Alzheimer's

disease. A recent work leverage saccade patterns for authentication [78]. We are inspired by these works. Our work differs in that we apply movement detection methods to adapt eye tracker's operations based on eye movement stages.

8 CONCLUSION AND FUTURE WORK

We designed and implemented a battery-free wearable eye tracker using low-cost NIR lights and photodiodes. While results from prototype experiments are promising, we recognize the limitations of our current study and summarize the future work. *First*, our current design focuses on indoor scenarios where ambient NIR light is relatively weak. We will extend the design to outdoor scenarios, by dealing with the strong NIR in sunlight, which can saturate NIR photodiodes and interfere with light sensing. We consider adding adaptive amplifiers to adjust the gain of amplifying signals. *Second*, tracking at 120 Hz, our system is unable to detect micro-saccades. We will seek to further boost the tracking rate while maintaining a similar energy budget. One method is to detect motion stages using less sensing data so that we can frequently turn off sensors to save energy. *Finally*, going beyond small-scale studies, we plan to conduct longer-term deployment with our prototype and examine various health monitoring applications. We will continue optimizing the circuit for a more miniaturized look, and consider its integration on regular glasses with various shapes.

REFERENCES

- [1] 2018. Gaze tracking on Microsoft HoloLens. <https://docs.microsoft.com/en-us/windows/mixed-reality/gaze>. (2018).
- [2] 2018. Harvard Database of Useful Biological Numbers. <http://bionumbers.hms.harvard.edu/bionumber.aspx?&id=100706&ver=4>. (2018).
- [3] 2018. IES Lighting Library. <https://www.ies.org/standards/ies-lighting-library/>. (2018).
- [4] 2018. PI NOIR CAMERA V2. <https://www.raspberrypi.org/products/pi-noir-camera-v2/>. (2018).
- [5] 2018. Pupil Labs. <https://docs.pupil-labs.com/>. (2018).
- [6] 2018. Recommended Light Levels. https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf. (2018).
- [7] 2018. Tobii Pro Glasses. <https://www.tobii.com/product-listing/tobii-pro-glasses-2/>. (2018).
- [8] 2018. Tobii Pro X2-60 eye tracker. <https://www.tobii.com/product-listing/tobii-pro-x2-60/>. (2018).
- [9] 2018. Vishay surface emitting technology. http://www.vishay.com/docs/49623/_pt0282.pdf. (2018).
- [10] Robert W Baloh, Andrew W Sills, Warren E Kumley, and Vicente Honrubia. 1975. Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology* 25, 11 (1975), 1065–1065.
- [11] F Behrens, M MacKeben, and W Schröder-Preikschat. 2010. An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior research methods* 42, 3 (2010), 701–708.
- [12] Inessa Bekerman, Paul Gottlieb, and Michael Vaiman. 2014. Variations in eyeball diameters of the healthy adults. *Journal of ophthalmology*

2014 (2014).

- [13] Anna Rita Bentivoglio, Susan B Bressman, Emanuele Cassetta, Donatella Carretta, Pietro Tonali, and Alberto Albanese. 1997. Analysis of blink rate patterns in normal subjects. *Movement Disorders* 12, 6 (1997), 1028–1034.
- [14] PR Bittencourt, P Wade, AT Smith, and A Richens. 1981. The relationship between peak velocity of saccadic eye movements and serum benzodiazepine concentration. *British journal of clinical pharmacology* 12, 4 (1981), 523–533.
- [15] James Brandt. 1991. Dictionary of Eye Terminology. (1991).
- [16] Katherine Breeden and Pat Hanrahan. 2017. Gaze Data for the Analysis of Attention in Feature Films. *ACM Transactions on Applied Perception (TAP)* 14, 4 (2017), 23.
- [17] A Buizza and R Schmid. 1986. Velocity characteristics of smooth pursuit eye movements to different patterns of target motion. *Experimental brain research* 63, 2 (1986), 395–401.
- [18] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2009. *Wearable EOG goggles: eye-based interaction in everyday environments*. ACM.
- [19] A. Bulling, J. A. Ward, H. Gellersen, and G. Troster. 2011. Eye Movement Analysis for Activity Recognition Using Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 4 (April 2011), 741–753.
- [20] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. 2008. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*. ACM, 96–103.
- [21] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 161–168.
- [22] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proc. of SIGKDD*.
- [23] Trevor J Crawford, Steve Higham, Ted Renvoize, Julie Patel, Mark Dale, Anur Suriya, and Sue Tetley. 2005. Inhibitory control of saccadic eye movements and cognitive impairment in Alzheimer’s disease. *Biological psychiatry* 57, 9 (2005), 1052–1060.
- [24] Edward Cutrell and Zhiwei Guan. 2007. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 407–416.
- [25] Andrew T Duchowski. 2007. Eye tracking methodology. *Theory and practice* 328 (2007).
- [26] Y. Ebisawa and S. i. Satoh. 1993. Effectiveness of pupil area detection technique using two light sources and image difference method. In *Proc. of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Societ.* 1268–1269.
- [27] Burkhard Fischer and E Ramsperger. 1984. Human express saccades: extremely short reaction times of goal directed eye movements. *Experimental Brain Research* 57, 1 (1984), 191–195.
- [28] RC Frecker, M Eizenman, and WJ MacLean. 1989. An evaluation of optical radiation hazards associated with infrared corneal. *CM-BEC15CCGB, Toronto, Ontario, Canada* (1989).
- [29] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
- [30] Jukka Häkkinen, Takashi Kawai, Jari Takatalo, Reiko Mitsuya, and Göte Nyman. 2010. What do people look at when they watch stereoscopic movies?. In *Stereoscopic Displays and Applications XXI*, Vol. 7524. International Society for Optics and Photonics, 75240E.
- [31] Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. 2012. Eye Tracker Data Quality: What It is and How to Measure It. In *Proc. of the Symposium on Eye Tracking Research and Applications*.
- [32] Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Stephen CF Chan, and Hong Va Leong. 2016. Building a personalized, auto-calibrating eye tracker from user interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5169–5179.
- [33] Edmund Burke Huey. 1908. *The psychology and pedagogy of reading*. The Macmillan Company.
- [34] Jukka Hyönä, Robert F Lorch Jr, and Johanna K Kaakinen. 2002. Individual differences in reading to summarize expository text: Evidence from eye fixation patterns. *Journal of Educational Psychology* 94, 1 (2002), 44.
- [35] William G Iacono, Margaret Moreau, Morton Beiser, Jonathan AE Fleming, and Tsung-Yi Lin. 1992. Smooth-pursuit eye tracking in first-episode psychotic patients and their relatives. *Journal of Abnormal Psychology* 101, 1 (1992), 104.
- [36] Kenji Irie, Bruce A Wilson, Richard D Jones, Philip J Bones, and Tim J Anderson. 2002. A laser-based eye-tracking system. *Behavior Research Methods, Instruments, & Computers* 34, 4 (2002), 561–572.
- [37] Olivia E. Kang, Katherine E. Huffer, and Thalia P. Wheatley. 2014. Pupil Dilation Dynamics Track Attention to High-Level Information. *PLOS ONE* 9, 8 (08 2014), 1–6.
- [38] Oleg V Komogortsev and Javed I Khan. 2007. Kalman filtering in the design of eye-gaze-guided computer interfaces. In *International Conference on Human-Computer Interaction*. Springer, 679–689.
- [39] Ravi Kothari and Jason L Mitchell. 1996. Detection of eye locations in unconstrained visual images. In *Image Processing, 1996. Proceedings., International Conference on*, Vol. 3. IEEE, 519–522.
- [40] Bartosz Kunka and Bozena Kostek. 2009. Non-intrusive infrared-free eye tracking method. In *Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA), 2009*. IEEE, 105–109.
- [41] Tianxing Li, Qiang Liu, and Xia Zhou. 2017. Ultra-Low Power Gaze Tracking for Virtual Reality. In *Proc. of SenSys*.
- [42] Tianxing Li and Xia Zhou. 2018. Battery-Free Eye Tracker on Glasses. In *Proc. of MobiCom*.
- [43] Simon P. Liversedge and John M. Findlay. 2000. Saccadic eye movements and cognition. *Trends in Cognitive Sciences* 4, 1 (2000), 6 – 14.
- [44] Addison Mayberry, Pan Hu, Benjamin Marlin, Christopher Salthouse, and Deepak Ganesan. 2014. iShadow: design of a wearable, real-time mobile gaze tracker. In *Proc. of MobiCom*.
- [45] Addison Mayberry, Yamin Tun, Pan Hu, Duncan Smith-Freedman, Deepak Ganesan, Benjamin M Marlin, and Christopher Salthouse. 2015. CIDER: Enabling robustness-power tradeoffs on a computational eyeglass. In *Proc. of MobiCom*.
- [46] Yair Morad, Hadas Lemberg, Nehemiah Yofe, and Yaron Dagan. 2000. Pupillography as an objective indicator of fatigue. *Current Eye Research* 21, 1 (2000), 535–542.
- [47] C.H Morimoto, D Koons, A Amir, and M Flickner. 2000. Pupil detection and tracking using multiple light sources. *Image and Vision Computing* 18, 4 (2000), 331 – 335.
- [48] Carlos Hitoshi Morimoto, Dave Koons, Arnon Amir, and Myron Flickner. 2000. Pupil detection and tracking using multiple light sources. *Image and vision computing* 18, 4 (2000), 331–335.
- [49] Tamami Nakano, Makoto Kato, Yusuke Morito, Seishi Itoi, and Shigeru Kitazawa. 2013. Blink-related momentary activation of the default mode network while viewing videos. *Proceedings of the National Academy of Sciences* 110, 2 (2013), 702–706.
- [50] Harald Niederreiter. 1988. Low-discrepancy and low-dispersion sequences. *Journal of number theory* 30, 1 (1988), 51–70.
- [51] P-H Niemenlehto. 2009. Constant false alarm rate detection of saccadic eye movements in electro-oculography. *Computer methods and*

- programs in biomedicine* 96, 2 (2009), 158–171.
- [52] Marcus Nyström and Kenneth Holmqvist. 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods* 42, 1 (2010), 188–204.
 - [53] Gillian A O’driscoll, Mark F Lenzenweger, and Philip S Holzman. 1998. Antisaccades and smooth pursuit eye tracking and schizotypy. *Archives of general psychiatry* 55, 9 (1998), 837–843.
 - [54] Takehiko Ohno, Naoki Mukawa, and Atsushi Yoshikawa. 2002. FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, 125–132.
 - [55] Sandra Oliveira, Friderike Oehler, Jesús San-Miguel-Ayanz, Andrea Camia, and José MC Pereira. 2012. Modeling spatial patterns of fire occurrence in Mediterranean Europe using Multiple Regression and Random Forest. *Forest Ecology and Management* 275 (2012), 117–129.
 - [56] Naoyuki Osaka. 1992. Size of saccade and fixation duration of eye movements during reading: Psychophysics of Japanese text processing. *JOSA A* 9, 1 (1992), 5–13.
 - [57] Alexandra Papoutsaki, James Laskey, and Jeff Huang. 2017. SearchGazer: Webcam Eye Tracking for Remote Studies of Web Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, 17–26.
 - [58] James L. Reilly, Rebekka Lencer, Jeffrey R. Bishop, Sarah Keedy, and John A. Sweeney. 2008. Pharmacological treatment effects on eye movement control. *Brain and Cognition* 68, 3 (2008), 415 – 435.
 - [59] JPH Reulen, JT Marcus, D Koops, FR De Vries, G Tiesinga, K Boshuizen, and JE Bos. 1988. Precise recording of eye movement: the IRIS technique Part 1. *Medical and Biological Engineering and Computing* 26, 1 (1988), 20–26.
 - [60] D. A. Robinson. 1963. A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field. *IEEE Transactions on Bio-medical Electronics* 10, 4 (Oct 1963), 137–145.
 - [61] Timothy A Salthouse and Cecil L Ellis. 1980. Determinants of eye-fixation duration. *The American journal of psychology* (1980), 207–234.
 - [62] Dario D. Salvucci. 1999. Inferring Intent in Eye-based Interfaces: Tracing Eye Movements with Process Models. In *Proc. of CHI*.
 - [63] Dario D. Salvucci. 1999. *Mapping Eye Movements to Cognitive Processes*. Ph.D. Dissertation. Carnegie Mellon University.
 - [64] Dario D Salvucci and Joseph H Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, 71–78.
 - [65] N Sarkar, B O’Hanlon, A Rohani, D Strathearn, G Lee, M Olfat, and RR Mansour. 2017. A resonant eye-tracking microsystem for velocity estimation of saccades and foveated rendering. In *Micro Electro Mechanical Systems (MEMS), 2017 IEEE 30th International Conference on*. IEEE, 304–307.
 - [66] Louis L Scharf and Cédric Demeure. 1991. *Statistical signal processing: detection, estimation, and time series analysis*. Vol. 63. Addison-Wesley Reading, MA.
 - [67] Jeroen BJ Smeets and Ignace TC Hooge. 2003. Nature of variability in saccades. *Journal of neurophysiology* 90, 1 (2003), 12–20.
 - [68] Mark B Spitzer and Jeffrey Jacobsen. 1994. Eye tracking system having an array of photodetectors aligned respectively with an array of pixels. (July 19 1994). US Patent 5,331,149.
 - [69] SR Research Ltd. 2005. EyeLink 1000 User Manual. (2005).
 - [70] Patrick Suppes. 1990. Eye-movement models for arithmetic and reading performance. *Eye movements and their role in visual and cognitive processes* 4 (1990), 455–477.
 - [71] Ambuj Varshney, Andreas Soleiman, Luca Mottola, and Thiemo Voigt. 2017. Battery-free Visible Light Sensing. In *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems*. ACM, 3–8.
 - [72] MéLodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2012. Wearable eye tracking for mental health monitoring. *Computer Communications* 35, 11 (2012), 1306–1311.
 - [73] RJM Vullers, Rob van Schaijk, Inge Doms, Chris Van Hoof, and R Mertens. 2009. Micropower energy harvesting. *Solid-State Electronics* 53, 7 (2009), 684–693.
 - [74] G. Waldman. 2002. *Introduction to Light: The Physics of Light, Vision, and Color*. Dover Publications.
 - [75] Bruce A Wilson, Richard D Jones, Philip J Bones, and Tim J Anderson. 1996. A flying-spot laser scanner for tracking eye movements. In *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, Vol. 1. IEEE, 146–147.
 - [76] Alfred L Yarbus. 1967. Eye movements during perception of complex objects. In *Eye movements and vision*. Springer, 171–211.
 - [77] Lan Zhang, Xiang-Yang Li, Wenchao Huang, Kebin Liu, Shuwei Zong, Xuesi Jian, Puchun Feng, Taeho Jung, and Yunhao Liu. 2014. It Starts with iGaze: Visual Attention Driven Networking with Smart Glasses. In *Proc. of MobiCom*.
 - [78] Yongtuo Zhang, Wen Hu, Weitao Xu, Chun Tung Chou, and Jiankun Hu. 2018. Continuous Authentication Using Eye Movement Response of Implicit Visual Stimuli. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 177.
 - [79] Jie Zhu and Jie Yang. 2002. Subpixel eye gaze tracking. In *Automatic face and gesture recognition, 2002. proceedings. fifth ieee international conference on*. IEEE, 131–136.